

Protocolo SSL

Secure Socket Layer: Secure Socket Layer (SSL) es un protocolo criptográfico para dar seguridad a la transmisión de datos. Este protocolo permite abrir conexiones seguras a través de una red, cifrando los datos intercambiados entre cliente y servidor mediante un algoritmo de cifrado simétrico. Para poder intercambiarse la clave de sesión utilizada entre cliente y servidor, se utiliza un algoritmo de cifrado de clave pública, típicamente RSA. Como algoritmo de hash se utilizan SHA-256, SHA1, MD5, etc. Para cada transacción de envío de datos se genera una clave de sesión distinta. La principal área de aplicación de las conexiones SSL es asegurar la comunicación entre el navegador y el servidor web. También se utiliza frecuentemente para asegurar la comunicación entre servidores.

SSL tiene como objetivos:

- Seguridad: mediante el cifrado de datos se garantiza que terceros no podrán tener acceso a la información cuando es enviada a través de la red.

- Integridad de los datos: la información enviada mediante una conexión puede ser validada en los extremos para comprobar que no ha sido alterada durante el camino.

- Autenticidad: nadie puede hacerse pasar por un sitio porque gracias a los algoritmos de encriptación se comprueba que los datos realmente han llegado al servidor que el cliente espera. La autenticidad permite evitar fraudes y ataques.

- Certificados digitales: Las conexiones SSL requieren que el servidor disponga de un certificado digital, el cual consiste en un archivo que identifica de modo único tanto a individuos como a servidores. El servidor debe autenticarse antes de establecer la sesión SSL.

PKI X.509 es un estándar UIT-T para infraestructuras de claves públicas. X.509 especifica, entre otras cosas, formatos estándar para certificados de claves públicas y un algoritmo de validación de la ruta de certificación. A partir de este estándar, se han desarrollado nuevos estándares que utilizan certificados X.509 en conexiones de correo seguro, comercio electrónico, etc

Los certificados X.509 contienen información relacionada con el usuario, la entidad emisora y el certificado en sí mismo, además de la firma digital del emisor. Los

certificados digitales son generalmente firmados por una autoridad de certificación (CA), la cual es fiable y permite garantizar la validez del certificado. Ejemplos: Thawtee, VeriSign, GeoTrust, RapidSSL.

SSL o Secure Socket Layer permite que, una vez esté habilitado, todas las comunicaciones entre el servidor y el cliente estén cifradas, de forma que terceros no puedan entender los datos que viajan del cliente al servidor y viceversa.

Los certificados tienen dos funciones principales en este proceso:

- Asegurar la identidad del servidor. Para que no haya posibilidad de suplantación por un tercero.
- Proporcionar las claves de cifrado. Aunque durante una sesión SSL la clave que se usa es simétrica (la clave para cifrar y descifrar es la misma), durante el proceso inicial de negociación de esa clave simétrica entre el navegador y el servidor, se utilizarán las claves asimétricas de los certificados para establecer un canal seguro por el que poder transmitir esa clave simétrica.

Uso de la herramienta keytool

Keytool es una herramienta que podemos encontrar tanto en el JDK como en el JRE de Sun Microsystems.

Primero generamos un par de claves (pública / privada) para nuestra organización. Este par de claves se guardan en un certificado autofirmado.

También crearemos nuestro almacén de claves (a partir de ahora nos referiremos al almacén de certificados como “keystore”), si es que todavía no estaba creado.

Este keystore será donde se guardará el certificado autofirmado con el par de claves (pública / privada).

Para crear el almacén:

```
keytool -genkey -alias tomcatw -keyalg RSA -keystore almacenw.jks -keysize 2048
```

“tomcatw” es el nombre del alias con el que hacemos referencia al par de claves creado.

“almacenw.jks” es el nombre del almacén.

Al crearlo solicita una clave, es la contraseña de acceso al almacén (en el ejemplo keypass: "pswxyz").

En nuestro caso el almacen está creado y la password es reed123.

Luego creamos la solicitud de certificado.

```
keytool -certreq -keyalg RSA -alias tomcatw -file certreq.csr -keystore almacenw.jks
```

-file: es el nombre del archivo donde se guarda la solicitud.

-alias: es el nombre con el que haremos referencia al par de claves creado.

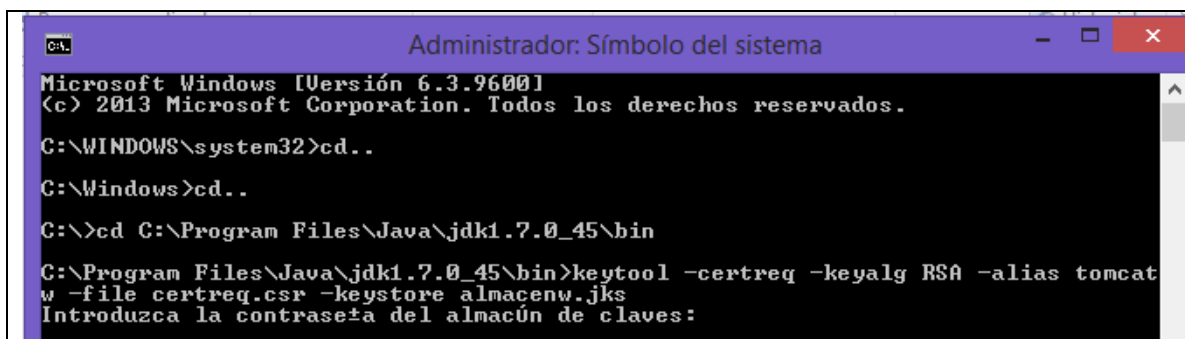


Figura 5: Herramienta keytool (ejecutar con privilegios de administrador).

Durante el proceso nos pedirá el nombre y apellidos (si estamos generando un certificado para aplicar SSL en un servidor Web de Internet, deberíamos poner el nombre DNS, si el servidor está en intranet deberíamos poner el nombre de la máquina), el nombre de la unidad organizativa, el nombre de la organización, el nombre de la ciudad, el nombre de la provincia, y el código de dos letras del país.

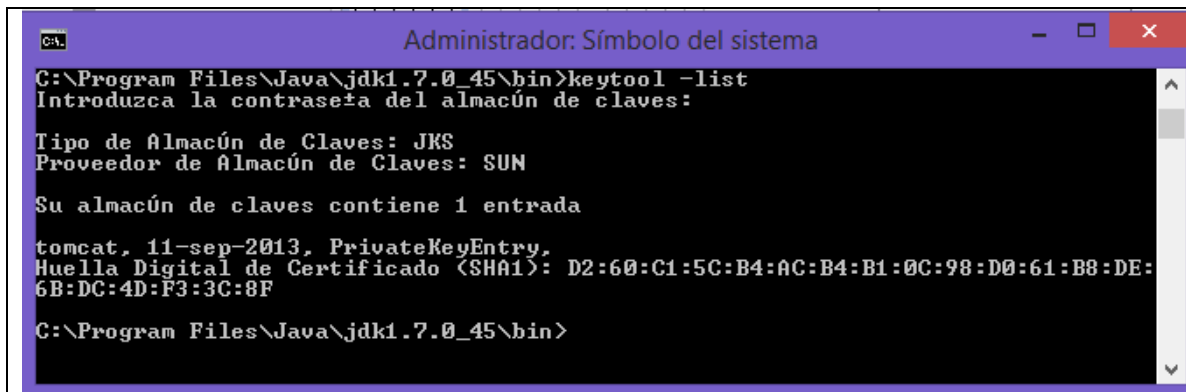


Figura 6: Keytool opción list permite explorar el almacén.

El archivo certreq.csr es enviado a la entidad que genera el certificado de confianza.

Nota: podemos “suplantar” a la CA. Lo que vamos a hacer es crearnos nuestra propia CA, y nos firmaremos a nosotros mismos la petición.

A esto no lo podemos hacer con keytool, así que tendremos que utilizar otra herramienta. Usaremos OpenSSL (<http://www.openssl.org/>). Open SSL es un conjunto de herramientas Open Source que implementan SSL v2/v3 y TLS v1, también se puede considerar como una librería criptográfica de propósito general. Podemos encontrar versiones tanto para Linux como para Windows.

La entidad (Thawte, GeoTrust, RapidSSL, VeriSign u otra) envía el certificado a través de un mail como archivo adjunto, o bien a través del envío de un vínculo.

La instalación del certificado se realiza en dos pasos ya que requiere la instalación de un certificado intermedio:



Nombre	Fecha de ...	Tipo	Tamaño
 dominio.crt	21/12/201...	Certificado de seguridad	2 KB
 intermedio.crt	21/12/201...	Certificado de seguridad	2 KB

Figura 7: Certificado de dominio y certificado intermedio.

Primero instalar el certificado intermedio usando la opción import de la herramienta keytool:

1) Haciendo doble click sobre el certificado se ve la estructura.

keytool -import -alias root -trustcacerts -file intermedio.crt -keystore almacenw.jks

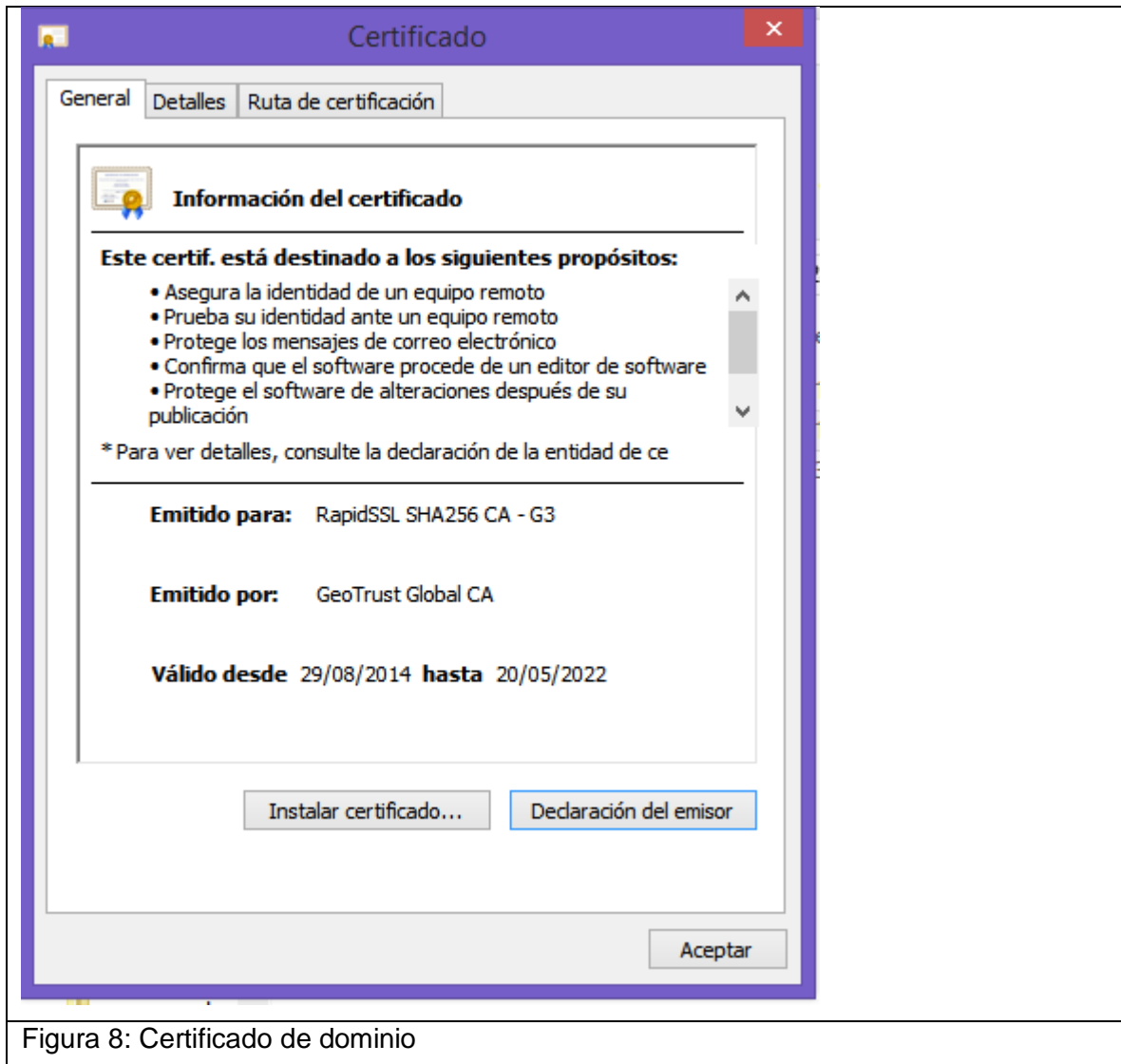


Figura 8: Certificado de dominio

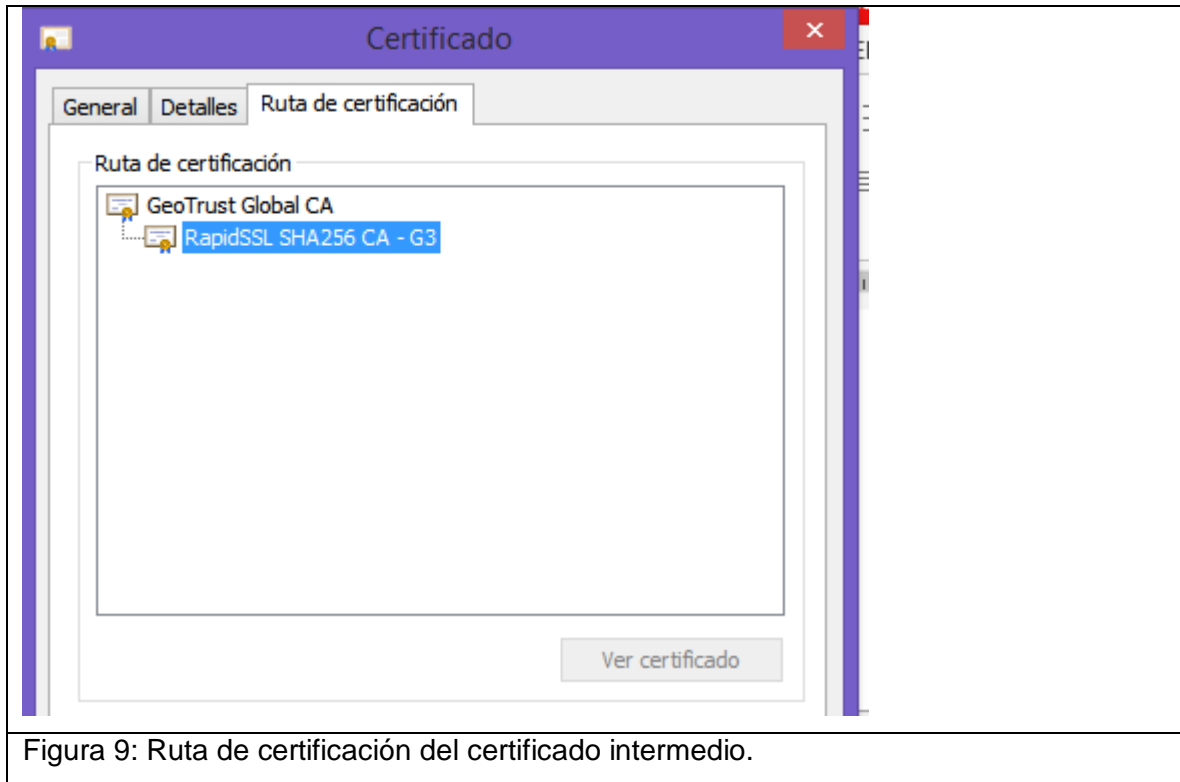


Figura 9: Ruta de certificación del certificado intermedio.

2) Importar el certificado del dominio.

```
keytool -import -alias tomcatw -trustcacerts -file dominio.crt -keystore almacenw.jks
```

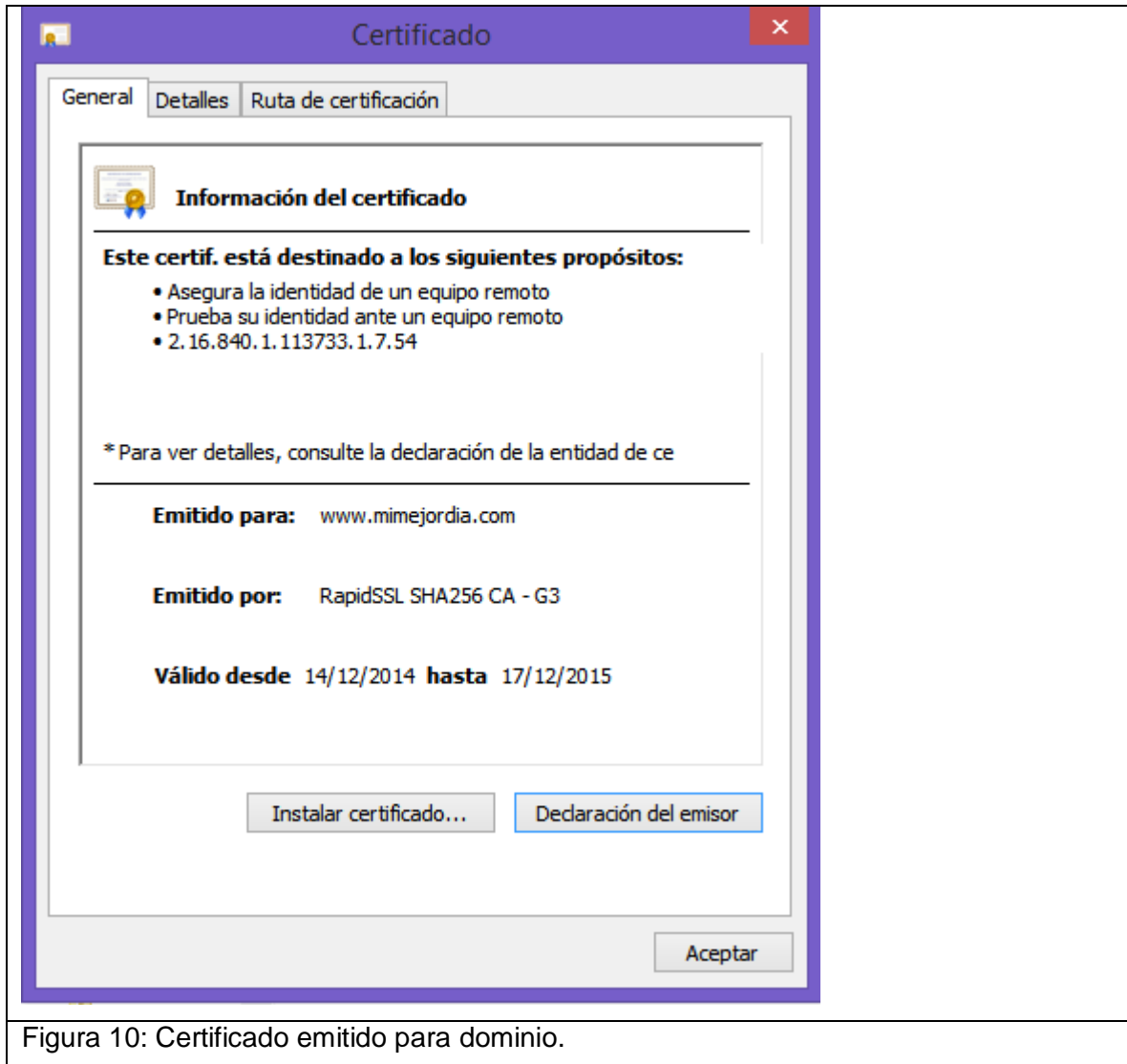


Figura 10: Certificado emitido para dominio.

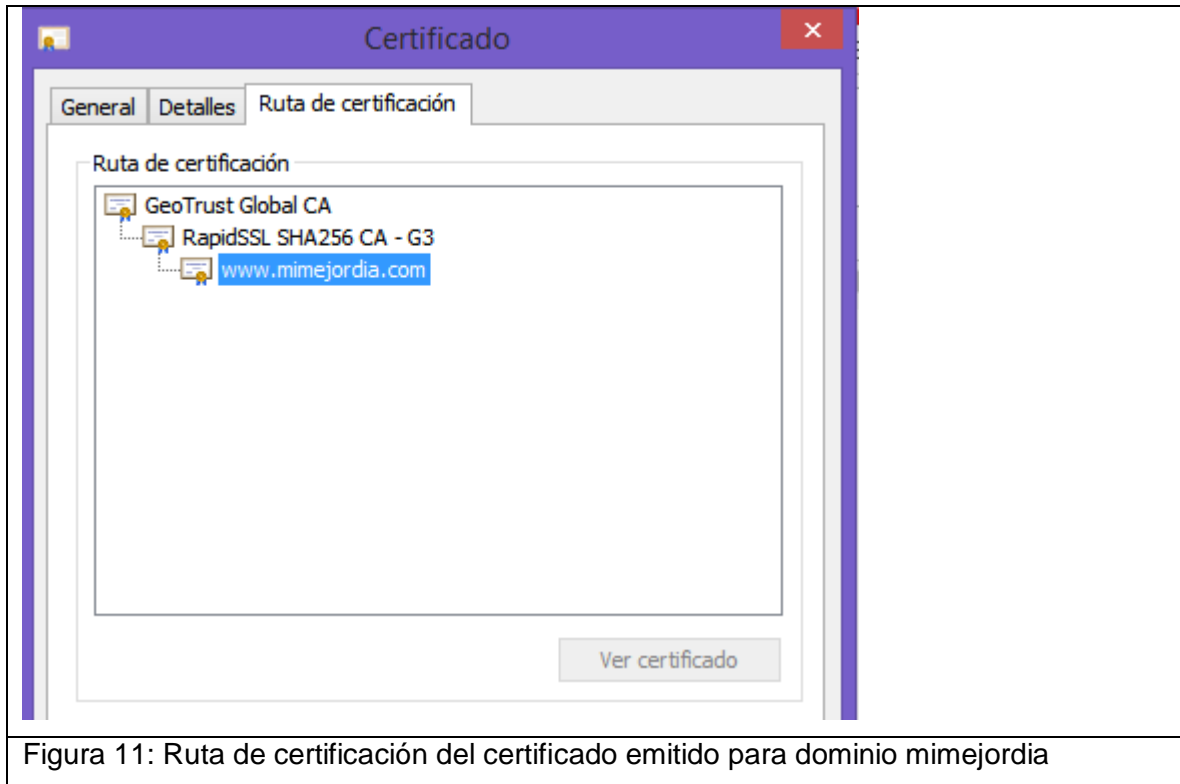


Figura 11: Ruta de certificación del certificado emitido para dominio mimejordia

3) Actualizar el server.xml

En el archivo server.xml del Apache Tomcat se especifica la ruta al almacén contenedor del certificado.

```
<Connector port="8443"
protocol="org.apache.coyote.http11.Http11NioProtocol"
SSLEnabled="true" maxThreads="150" scheme="https"
secure="true" clientAuth="false" sslProtocol="TLS"
keystoreFile="C:\Program Files\Apache Software Foundation\Tomcat 7.0_Tomcat7.0.53\GeoTrust\almacenw.jks"
keystorePass="pswxyz" URIEncoding="UTF-8" useBodyEncodingForURI="true"
/>
```

Figura 12: Configuración del certificado en el archivo server.xml

Generalmente las empresas que venden los certificados también ofrecen mecanismos para validarlos.

Enter your Web Server's domain name:

Enter your port (443 is default for SSL):

Status: Successful

www.mimejordia.com is successfully secured by an SSL certificate.
The following certificates are correctly installed:

-----Certificate 1-----
--Issued To--
Organizational Unit: Domain Control Validated - RapidSSL(R)
Organizational Unit 2: See www.rapidssl.com/resources/cps (c)14
Organizational Unit 3: GT61060926
Common Name: www.mimejordia.com

--Issued By--
Organization: GeoTrust Inc.
Common Name: RapidSSL SHA256 CA - G3
Country: US

Valid from Sun Dec 14 08:08:30 ART 2014 to Thu Dec 17 10:23:33 ART 2015
Serial Number (hex): 01176a
Signature Algorithm: SHA256withRSA
Key Size: 2048 bits
SANs: www.mimejordia.com, mimejordia.com

-----Certificate 2-----
--Issued To--
Organization: GeoTrust Inc.
Common Name: RapidSSL SHA256 CA - G3
Country: US

--Issued By--
Organization: GeoTrust Inc.
Common Name: GeoTrust Global CA
Country: US

--Issued By--
Organization: GeoTrust Inc.
Common Name: GeoTrust Global CA

Country: US

Valid from Fri Aug 29 18:39:32 ART 2014 to Fri May 20 18:39:32 ART 2022

Serial Number (hex): 023a77

Signature Algorithm: SHA256withRSA

Key Size: 2048 bits

-----Certificate 3-----

--Issued To--

Organization: GeoTrust Inc.

Common Name: GeoTrust Global CA

Country: US

--Issued By--

Organization: GeoTrust Inc.

Common Name: GeoTrust Global CA

Country: US

Valid from Tue May 21 01:00:00 ART 2002 to Sat May 21 01:00:00 ART 2022

Serial Number (hex): 023456

Signature Algorithm: SHA1withRSA

Key Size: 2048 bits

Figura 13: Test para validar el certificado provisto por Rapid SSL

Nota: Generalmente para ejecutar el validador es preciso reducir la seguridad de la consola java.